Inventor(s): Clark D. JEFFRIES, Charles S. LINGAFELT and Norman C. STROLE

# METHOD AND SYSTEM FOR CONTROLLING DATAFLOW TO A CENTRAL SYSTEM FROM DISTRIBUTED SYSTEMS

## FIELD OF THE INVENTION

The present invention relates to computer networks, and more particularly to a method and system for controlling dataflow from distributed systems to a central system.

## BACKGROUND OF THE INVENTION

Figure 1 depicts a high-level block diagram of a conventional computer system 10 that includes a conventional central system 12 and multiple conventional distributed systems 14, 16, 18, and 20. The conventional distributed systems 14, 16, 18, and 20 can be viewed as communicating with the conventional central system 12 via pipes 13, 15, 17, and 19, respectively. Examples of such conventional computer systems 10 include intrusion detection systems in which the conventional central system 12 detects intrusions based upon distributed clients 14, 16, 18, and 20 that detect input to the conventional computer system 10. Such conventional distributed systems 14, 16, 18, and 20 then alert the conventional central system 10 of the intrusions by providing data packets via pipes 13, 15, 17, and 19,

respectively. In the event of an attack, the conventional central computer system 12 may be able to prevent failure of the conventional computer system 10 due to such attacks. Other examples of such systems includes bridge computing or other systems which employ distributed systems, such as the conventional clients 14, 16, 18, and 20 that communicate directly to a conventional central system 12.

Although the system 10 functions, one of ordinary skill in the art will readily realize that the flow of data packets from the multiple conventional distributed systems 14, 16, 18, and 20 through the conventional pipes 13, 15, 17, and 19 is unregulated. In particular, it is possible for the traffic through the conventional pipes 13, 15, 17, and 19 to be sufficiently high that the conventional central system 10 is overwhelmed. For example, if the conventional computer system 10 is being attacked by a flood of packets being denied service (a denial of service attack), then one or more of the conventional distributed systems 14, 16, 18, and 20 may overwhelm the conventional central system 12 by providing an alert for each denial of service. Similarly, if the conventional distributed systems 14, 16, 18, and 20 are simply conventional clients linked to the conventional central system 12 and there is some interruption of service or other accident, one or more of the conventional distributed systems 14, 16, 18, and 20 may provide the conventional central system 12 with multiple notifications of the accident via the conventional pipes 13, 15, 17, and 19, respectively. As a result, the conventional system 10 may fail.

Although there may be many conventional methods for preventing a failure of the conventional system 10 despite unregulated traffic from the conventional distributed systems 14, 16, 18, and 20. For example, a maximum threshold may be placed on traffic from one or more of the pipes 13, 15, 17, and 19. If traffic though the pipes 13, 15, 17, or 19 exceeds

this threshold, then the data packets are discarded. However, discarding of packets is undesirable if the excessive flow of data packets is not due to an attack. Furthermore, in some instances, the allocation of resources such as bandwidth might be changed to account for high traffic to the conventional central system 12 from some portion of the distributed systems 14, 16, 18, and 20.

Accordingly, what is needed is a system and method for better controlling traffic from distributed systems to a central system. The present invention addresses such a need.

## SUMMARY OF THE INVENTION

The present invention provides a method and system for controlling a plurality of pipes in a computer system including at least one central system. The plurality of pipes provides traffic from a plurality of distributed systems. The method and system comprise providing a first plurality of data packets from a pipe of the plurality of pipes to a fast path or a slow path during a time interval such that none of the first plurality of data packets is dropped. The first plurality of data packets arrive in a time interval. The fast path includes a fast storage, while the slow path includes a bulk storage. The method and system also comprise providing a second plurality of data packets from the fast storage or the bulk storage to the central system in a first in first out order during the time interval.

According to the system and method disclosed herein, the present invention regulates traffic from the distributed system to the central system.

## BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a conventional system that provides traffic from

distributed systems to a central system.

Figure 2 is a high-level block diagram of one embodiment of a system in accordance with the present invention for controlling traffic from a plurality of distributed systems to a central system.

Figure 3 is a high-level flow chart of one embodiment of a method in accordance with the present invention for controlling traffic from a plurality of distributed systems to a central system.

Figure 4 is a more detailed flow chart of one embodiment of a method in accordance with the present invention for controlling traffic from a plurality of distributed systems to a central system.

## DETAILED DESCRIPTION OF THE INVENTION

The present invention relates to an improvement in control of traffic in computer networks. The following description is presented to enable one of ordinary skill in the art to make and use the invention and is provided in the context of a patent application and its requirements. Various modifications to the preferred embodiment will be readily apparent to those skilled in the art and the generic principles herein may be applied to other embodiments. Thus, the present invention is not intended to be limited to the embodiment shown, but is to be accorded the widest scope consistent with the principles and features described herein.

The present invention provides a method and system for controlling a plurality of pipes in a computer system including at least one central system. The plurality of pipes provides traffic from a plurality of distributed systems. The method and system comprise providing a first plurality of data packets from a pipe of the plurality of pipes to a fast path or

a slow path during a time interval such that none of the first plurality of data packets is dropped. The first plurality of data packets arrive in a time interval. The fast path includes a fast storage, while the slow path includes a bulk storage. The method and system also comprise providing a second plurality of data packets from the fast storage or the bulk storage to the central system in a first in first out order during the time interval.

The present invention will be described in terms of a particular system and particular components. However, one of ordinary skill in the art will readily recognize that this method and system will operate effectively for other components in a computer network. For example, the present invention is described in the context of a single central system and a network processor. However, multiple central systems and other components capable of providing analogous functions might be used.

To more particularly illustrate the method and system in accordance with the present invention, refer now to Figure 2, depicting a high-level block diagram of one embodiment of a system 100 in accordance with the present invention for controlling traffic from a plurality of distributed systems to a central system. The system 100 is depicted as being used in the computer system 10' that is analogous to the conventional system 10 depicted in Figure 1. Referring back to Figure 2, the computer system 10' includes a central system 12', distributed systems 14', 16', 18' and 20', as well as the corresponding pipes 13', 15', 17', and 19', respectively. The system 100 includes a flow regulator 102, a fast path 104 including fast storage 106, a slow path 108 including bulk storage 110, and optional medium paths 112 including optional medium storage 114. In addition, an output storage 116, which is preferably a FIFO output queue, is shown.

The flow regulator 102 preferably includes a memory 103, such as a queue, into

which the data packets from the pipes 13', 15', 17', and 19' flow. The flow regulator 102 also includes a processor 101. The processor 101 examines the data packets from each pipe and determines into which path 104, 108, or 112, the data packets are to be transmitted. In a preferred embodiment, the flow regulator 102 contains a network processor 101. Network processors are preferred because such devices have already gained general use in controlling the flow of data packets in networks. Furthermore, network processors can be programmable. The programmability of the network processor 101 allows the flow regulator 102 to adapt to different conditions and applications using changes in the software controlling the network processor 101. Consequently, the system 100 is made more flexible and easy to use. In a preferred embodiment, the network processor 101 preferably selects the path 104, 108, or 112 such that none of the data packets entering the system 100 are dropped. The network processor 101 may operate based on a particular time interval for each pipe 13', 15', 17', and 19'. Thus, in a preferred embodiment, all packets from a particular pipe in the time interval for that pipe are classified in a like manner.

In one embodiment, the fast storage 106 includes a fast memory (e.g. a fast queue) that is relatively small and generally more expensive to manufacture. Similarly, the bulk storage 110 is relatively large, slower, and less expensive. Note that in other embodiments, more than one fast storage 106 and more than one bulk storage 110 could be used. The medium storage(s) preferably include memories with capabilities between the bulk storage and the fast storage. In one embodiment, only the fast path 104 and the slow path 108 are used. Consequently, the present invention is primarily described in the context of the fast path 104 and the slow path 108. As discussed above, the regulator 102 classifies data packets from each of the pipes 13', 15', 17' and 21'; selects the appropriate path 104, 108, or

112; and provides packets from the pipes 13', 15', 17', and 19 to the selected path such that no packets are dropped. Packets are provided from the fast path 104, the slow path 108 and the medium path(s) 112 in a first-in-first-out order.

Figure 3 is a high-level flow chart of one embodiment of a method 200 in accordance with the present invention for controlling traffic from a plurality of distributed systems to a central system. The method 200 is preferably performed by the system 100 and for the central system 12', pipes 13', 15', 17', and 19' and the distributed systems 14', 16', 18', and 20'. The method 200 will, therefore, be described in the context of the system 100 and other components depicted in Figure 2. Referring to Figures 2 and 3, the method 200 is preferably performed for each pipe 13', 15', 17', and 19'. However, for clarity, the method 200 is described in the context of the pipe 13'. A first plurality of data packets arrive from the pipe 13' in a time interval that is preferably specific to the pipe 13'. Stated differently, the time interval may be different for each pipe 13', 15', 17', and 19'. The first plurality of data packets from the pipe 13' is provided to the appropriate one of the paths 104, 108, or 112 during the time interval for the pipe 13' such that none of the first plurality of data packets is dropped, via step 202. In a preferred embodiment, only the fast path 104 and the slow path 108 are used. In such an embodiment, step 202 provides the first plurality of data packets to the fast path 104 or the slow path 108. The flow regulator 102 preferably determines to which path 104, 108, or 112, each data packet belongs in step 202 and forwards the packet to the appropriate path 104, 108, or 112. Also in a preferred embodiment, each of the first plurality of data packets is treated similarly. Thus, if one data packet in the time interval is provided to the fast path 104, then all data packets for the pipe 13' in the time interval are provided to the fast path 104.

During the time interval, a second plurality of data packets already in the paths 104, 108, and 112 are also provided from the paths 104, 108, and 112 to the central system 12' such that the data packets arrive at the central system 12' in the order received by the system 100, via step 204. Thus, in a preferred embodiment, the data packets are provided from the fast storage 104 or the bulk storage 110 to the output storage 116 and then the central system 12' in a first in first out order during the time interval. The method 200 is preferably repeated for each pipe 13', 15', 17' and 19' over each time interval for that pipe 13', 15', 17' and 19'.

Using the method 200, the system 100 can control traffic provided to the central system 12' from the distributed systems 14', 16', 18', and 20' through the pipes 13', 15', 17' and 19'. Because the data path 104, 108, or 112 is selected such that no packets are dropped, slower paths, such as the slow path 108, having greater storage capacity are used at times of higher congestion. Because the slow path 108 can be selected in times of greater congestion, the central system 12' may not be overwhelmed. In addition, the fast storage 106 does not overflow. The bulk storage 110 is also preferably large enough to avoid overflows. Consequently, no packets are lost. Instead, all packets may be provided to the central system 12' at some point, even for times during very high congestion. Moreover, in a preferred embodiment, selection of the slow path 108 in the step 202 is limited. Instead, the more efficient fast path 104 is generally used. Preferably, the fast storage 106 is used as much as possible. Data packets are preferably shunted to the slow path for the pipe 13' only during times of congestion for the pipe 13'. Consequently, performance of the system 10' is not compromised. Furthermore, the data packets are provided from the fast path 104 and the slow path 108 in order.

Figure 4 is a more detailed flow chart of a preferred embodiment of a method 210 in accordance with the present invention for controlling traffic from a plurality of distributed systems to a central system. The method 210 is preferably performed by the system 100 and for the central system 12', pipes 13', 15', 17', and 19' and the distributed systems 14', 16', 18', and 20'. The method 210 will, therefore, be described in the context of the system 100 and other components depicted in Figure 2. Referring to Figures 2 and 4, the method 210 is preferably performed for each pipe 13', 15', 17', and 19'. For clarity, the method 210 is described in the context of the pipe i, which could be any of the pipes 13', 15', 17', and 19'.

Using the method 210, the traffic from the pipe i to the central system 12' is controlled in increments of time equal to the time interval. The method 210 preferably commences after the time interval for the pipe i is determined. The time interval for pipe i, $DT_i$, depends upon the maximum possible arrival rate (data packets per time), $A_i$, for pipe i and the capacity, $C_i$, of the fast path 104 for pipe i. In a preferred embodiment, the storage capacity for the fast path 104 is preferably the capacity of the fast storage 106. For the pipe i, the time interval, $DT_i$ is not greater than and proportional to the storage capacity of the fast path 104 for the pipe i divided by the maximum possible arrival rate for the pipe i, or $C_i/A_i$. Stated differently, $DT_i$ is $F_i$ $(C_i/A_i)$, where $F_i$ is a factor for the $i^{th}$ pipe that is less than or equal to one. In a preferred embodiment, $DT_i$ is (1/8) $(C_i/A_i)$. In an alternate embodiment, a factor other than 1/8 could be used. For example, a factor of up to at least approximately ½ can be used. In general, the larger the time interval, the easier the method 210 is computationally, but the more likely that the fast path storage 106 will overflow. For smaller time intervals, it is less likely that the fast path storage 106 will overflow, but the method 210 is more computationally intensive. Thus, the appropriate factor can be selected

for the pipe i. In a preferred embodiment, the time interval is set prior to the method 210 commencing. However, in an alternate embodiment, the time interval may be set as part of the method 210. In such an embodiment, the time interval may be adjusted throughout operation of the system 100.

For managing traffic for a current interval, it is determined whether the occupation $Q_i$ of the fast storage (or fast queue) 106 is greater than a particular threshold, via step 212. The threshold for the fast storage 106 for pipe i, $T_i$, may be set at a particular fixed value. In a preferred embodiment, the threshold is set such that the fast path storage 106 will not overflow. Consequently, the threshold is preferably set such that the threshold plus the maximum possible input rate for pipe i multiplied by the time interval is less than the storage capacity of the fast path storage 106 for pipe i. Stated differently, the threshold is set such that $T_i + A_i*DT_i < C_i$. In a preferred embodiment, the determination in step 212 is based on values for a previous time interval. In other words, at time T, step 212 determines whether the occupation for pipe i for a previous interval $Q_i(T - DT_i)$ is less than $T_i$.

If the occupation of the fast storage 106 for pipe i is greater than the threshold for pipe i, then the packets for the current time interval are provided to the slow path 108, via step 214. In a preferred embodiment, step 214 is performed by setting a transmission signal for the pipe i to a value corresponding to the slow path 108, and forwarding the data packets input for pipe i during the time interval based upon the transmission signal. The transmission signal for pipe i, $S_i$, is preferably a binary value. At one value, the transmission signal corresponds to the fast path 104. At the other value, the transmission signal corresponds to the slow path. In a preferred embodiment, $S_i(T) = 1$ corresponds to the fast path 104, while $S_i(T) = 0$ corresponds to the slow path 108. Thus, when $S_i(T) = 1$, the data

packets provided from pipe i from the time T to the time T + $DT_i$ are provided to the fast path 104. Similarly, when $S_i(T) = 0$, the data packets provided from pipe i from the time T to the time T + $DT_i$ are provided to the slow path 108. Consequently, step 214 preferably includes setting $S_i(T)$ to zero and forwarding the packets arriving during the current time interval based on the zero value of $S_i(T)$.

If it is determined in step 212 that the occupation of the fast path storage 106 for pipe i does not exceed the threshold, then it is determined whether the slow path storage 110 for pipe i is empty, via step 216. The occupation of the slow path storage 106 for pipe i is given by $B_i$. When $B_i$ is zero, no packets for pipe i are contained in the slow path storage 106. In a preferred embodiment, the determination in step 216 is based on values for a previous time interval. In other words, at time T, step 216 determines whether the occupation of the slow path storage 110 for pipe i for a previous interval $B_i(T - DT_i)$ is zero.

If it is determined in step 216 that the occupation of the slow path storage 110 is zero, then the packets for the current time interval are provided to the fast path 104, via step 218. In a preferred embodiment, step 218 is performed by setting the transmission signal for the pipe i to a value corresponding to the fast path 104, and forwarding the data packets input for pipe i during the time interval based upon the transmission signal. Consequently, step 218 preferably includes setting $S_i(T)$ to one and forwarding the packets arriving during the current time interval based on the one value of $S_i(T)$.

If it is determined in step 216 that the occupation of the slow path storage 110 for pipe i is not zero, then it is determined whether the packets were being transmitted to the slow path, via step 220. Step 220 preferably includes determining whether the transmission signal is set such that data packets are sent to the slow path. In a preferred embodiment, the

determination in step 220 is based on values for a previous time interval. In other words, at time T, step 220 determines whether the transmission signal, $S_i(T - DT_i)$ was set such that the data packets travel to the slow path (e.g. $S_i(T - DT_i) = 0$).

If it is determined in step 220 that the transmission signal is set such that data packets travel to the slow path 108, then the packets for the current time interval are provided to the slow path 108, via step 222. In a preferred embodiment, step 222 is performed by setting the transmission signal for the pipe i to a value corresponding to the slow path 108, and forwarding the data packets input for pipe i during the time interval based upon the transmission signal. Consequently, step 218 preferably includes setting $S_i(T)$ to a zero and forwarding the packets arriving during the current time interval based on the one value of $S_i(T)$. If it is a determined in step 220 that the transmission signal is not set so that data packets travel to the slow path 108, then packets for the current time interval are provided to the fast path 104, via step 224. In parallel to steps 212 through 224, the data packets in the fast storage 108 and the slow storage 110 are provided to the central system 12' in a first-in-first-out order, via step 226. Once the current time interval expires, steps 212-228 are then repeated for the next time interval, via step 230. In repeating these steps, the next interval becomes the current interval (T goes to T + DT), and the current interval is the past interval (T – DT goes to T).

Thus, using the method 210, the system 100 can control traffic provided to the central system 12' from the distributed systems 14', 16', 18', and 20' through the pipes 13', 15', 17' and 19'. Because the data path 104, 108, or 112 is selected such that no packets are dropped, slower paths, such as the slow path 108, having greater storage capacity are used at times of higher congestion. Because the slow path 108 can be selected in times of greater

congestion, the central system 12' may not be overwhelmed. Moreover, in a preferred embodiment, selection of the slow path 108 in the step 202 is limited. Instead, the more efficient fast path 102 is generally used. Data packets are preferably shunted to the slow path for the pipe 13', 15', 17', and 19' only during times of congestion for the pipe 13', 15', 17', and 19'. Consequently, performance of the system 10' is not compromised.

A method and system has been disclosed for controlling traffic from distributed systems to a central system. Software written according to the present invention is to be stored in some form of computer-readable medium, such as memory, CD-ROM or transmitted over a network, and executed by a processor. Alternatively, some of all of the present invention could be implemented in hardware. Although the present invention has been described in accordance with the embodiments shown, one of ordinary skill in the art will readily recognize that there could be variations to the embodiments and those variations would be within the spirit and scope of the present invention. Accordingly, many modifications may be made by one of ordinary skill in the art without departing from the spirit and scope of the appended claims.